Decision Preparation: Multi-Dimensional Access Decisions & Edge Policy Distribution

Context

Designing a real-time access decision system for multi-dimensional factors (user, time, location, booking, recent activity) on constrained edge devices. The goal: low latency, high reliability, and minimal compute while securely managing policy and credential updates.

1. Problem Definition

Current system uses complex SQL joins across user, booking, and time tables—too slow for real-time edge scenarios. We need declarative, low-latency decisions executed locally with dynamic, time-limited credentials.

2. Core Insight

This is a pattern-matching and policy evaluation problem, not a database query problem. The architecture should use declarative policies, attributes, relationships, and contextual events (CEP).

3. Architectural Direction

Two viable policy languages:

Cedar (Rust-native): clear syntax, small footprint, and fast edge runtime.

Rego (OPA, WASM): flexible and widely used, supports partial evaluation for compact edge binaries.

4. Edge Policy Architecture

• PEP (lock/controller) gathers attributes and requests decision. • PDP (gateway/lock) evaluates policy in Cedar or Rego→WASM. • CEP layer (Redis Streams or flash cache) stores 'last_open' and recent events. • Attribute cache replaces heavy SQL joins.

5. Policy Example (Cedar)

permit(principal, action, resource) when { principal.clearance >= resource.required_clearance && resource.required_group in principal.groups && withinTimeAny(context.time, resource.allow_hours) && withinGeofence(context.location, resource.geofence) && noImpossibleTravel(context.last_open, context.location, context.time) };

6. Booking-Based Access Window

Example: booking 12:00–13:00. Credential valid 11:50–13:20. Delivered as signed CBOR/JWT capability token with 30–40 min TTL. Token fields: sub, aud, res, act, nbf, exp, sig.

7. Communication Layer

• **Thread/Matter**: ideal for battery locks (low power, mesh reliability) • **Wi-Fi**: for mains-powered controllers • **BLE/NFC**: offline fallback Tokens ≤256 bytes (CBOR+Ed25519) suitable for all transports.

8. Security & Reliability

Short-lived credentials, key rotation, replay protection (jti), authenticated time sync, and offline-capable fallback combine for secure and reliable edge access control.

9. Decision Summary

- **Declarative language:** Cedar (Rust-native)
- **Alternative:** Rego → WASM (OPA)
- **Policy model:** ABAC/ReBAC + CEP context
- **Runtime footprint:** Rust binary or WASM module
- **Update model:** Push short-lived capability tokens
- **Transport:** Thread/Matter preferred, Wi-Fi fallback
- **State storage:** Redis or flash key-value cache
- **Revocation:** Time-based expiry or key rotation

10. Next Steps

1. Prototype Cedar & Rego→WASM policies 2. Benchmark PDP latency on MCU & gateway 3. Define capability token schema & signing service 4. Implement Thread/Matter push + BLE fallback 5. Integrate Redis CEP for 'last_open' 6. Establish CI tests for policy correctness

Executive Summary

This architecture replaces SQL-based authorization with declarative, edge-native policies. It shifts from pushing rules to pushing short-lived credentials, ensuring low latency, high reliability, and strong security in an ambient, intelligent access ecosystem.